



**BOĞAZİÇİ UNIVERSITY**  
**SOFTWARE ENGINEERING DEPARTMENT**

**SWE 573.01-SOFTWARE DEVELOPMENT PRACTICE**  
**MVP Milestone 2 Report**

Ahmet DOLMA  
ID: 2022719210

Date: 04.29.2024

## Table of Contents

1.	Important Links .....	3
2.	What is Done and Status.....	3
3.	Updated Class Diagram .....	5
4.	Explanation of What is Being Done and What has been Done How.....	5
4.1.	Security Configuration - Link .....	5
4.2.	User Registration.....	6
4.3.	Basic Structure of the Project .....	7
4.4.	Issues.....	7

## 1. Important Links

**Github Link:** <https://github.com/AhmetZ06/SWE-573---Ahmet-Dolma->

**Deployment Link:** <https://swe-573-ahmet-dolma.onrender.com/login>

### Example User Login Info

You can register from our Register but if you want existing users you can use the followings

User 1 – [can@can.com](mailto:can@can.com) – Password – can

### Instructions

**Just push the link and do the followings**

## 2. What is Done and Status

For the application status, the users can register, login, create and show available communities. Right now, default post template is ready but I have encountered a problem with post creation. Most of the classes for the models are ready but as the project goes on, I change them accordingly if any problem is encountered. Right now, my services and controllers for authentication, community creation and showing communities, also for security purposes adding and authenticating users created.

- ❖ Right now my application shows create post template to only admins of that community
- ❖ If an unauthorized user tries to access, it redirects to the login page
- ❖ Users can login
- ❖ Users can register
- ❖ Users can create communities
- ❖ Users can show available communities
- ❖ Users can enter community specific pages

I have encountered some problems and tag them as bugs in my repository issues.

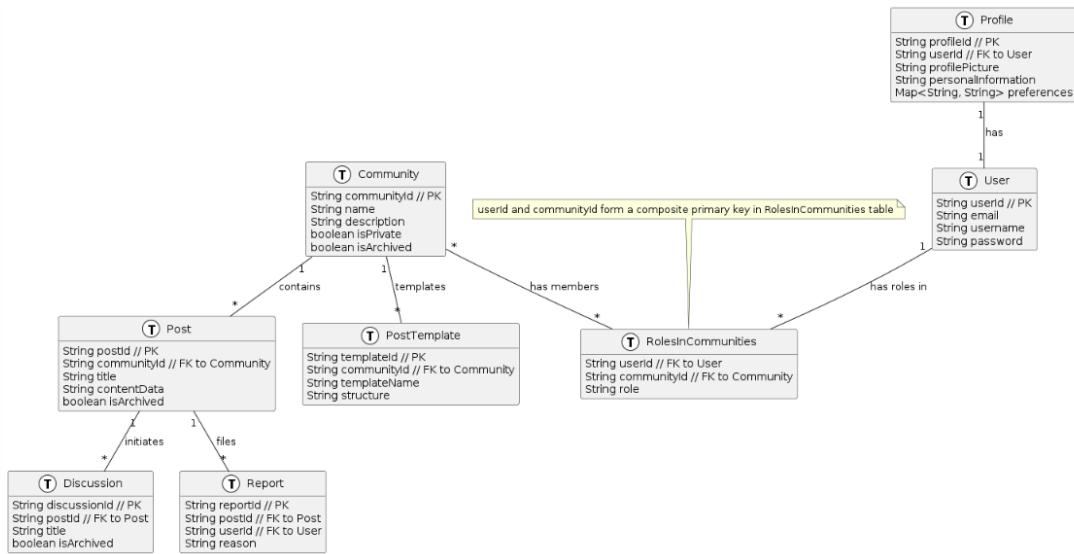
REQUIREMENT	DESCRIPTION	Completed	In Progress
F1. Registration	The system shall allow users to register an account.		-
F2. Profile Creation	The system shall prompt users to create a profile after registration.	-	
F3. Profile Customization	The system shall enable users to customize their profiles.	-	
F4. Information Management	The system shall provide users with the ability to manage their personal information.	-	
F5. Functional Requirement: Reporting System	The platform shall provide a reporting system, enabling users to report inappropriate content, behavior, or any violations of community guidelines.	-	

F6. Functional Requirement: User Follow System	The platform shall allow users to follow other users, enabling them to receive updates or notifications about the activities, posts, or contributions of the users they follow.	▪	
<b>Community Discovery and Membership</b>			
F7. Community Exploration	The system shall offer tools for exploring public communities.	▪	
F8. Membership Requests	The system shall allow users to request membership in private communities.	▪	
F9. Community Joining	The system shall enable users to join public communities.		▪
<b>Content Creation and Interaction</b>			
F10. Content Creation	The system shall allow community members to create content within communities.		▪
F11. Discussions and Insights	The system shall allow community members to engage in community discussions.	▪	
F12. Content Interaction	The system shall facilitate community members interaction with community content.	▪	
<b>Role-Specific Operations for Community Moderators</b>			
F13. Membership Management	The system shall allow moderators to manage community memberships.	▪	
F14. Content Moderation	The system shall enable moderators to control the content within their community.	▪	
F15. Discussion Moderation	The system shall support moderators in overseeing community discussions.	▪	
F16. Guideline Publication	The system shall allow moderators to publish community guidelines.	▪	
<b>Role-Specific Operations for Community Creators</b>			
F17. Moderator Assignment	The system shall permit community creators to assign moderator roles.	▪	
F18. Discussion Archiving	The system shall enable creators to archive discussions.	▪	
F19. Ownership Transfer	The system shall facilitate the transfer of community ownership.	▪	
F20. Owner Assignment	The system shall allow creators to assign new community owners.	▪	
<b>System Operations for User Registration and Profile Management</b>			
F21. System Registration Notification Operation	The system shall automatically send a verification email upon user registration.	▪	
F22. System Content Creation Notification Operation	The system shall notify the community owner and moderators when new content is posted.	▪	
<b>System Operations for Role-Specific Actions</b>			
F23. System Moderator Assignment Notification Operation	The system shall notify users when they are assigned as moderators by a community creator.	▪	

F24. System Ownership Transfer Notification Operation	The system shall notify members when the ownership of the community is transferred.	-	
F25. System Discussion Archiving Operation	The system shall allow community creators to archive discussions, rendering them read-only.	-	
<b>System Management Operations</b>			
F26. User Authentication System Operation	The system shall authenticate all users to secure user accounts and unauthorized access.		-
F27. System Maintenance Notification Operation	The system shall inform users of maintenance activities during periods when the platform is not accessible.	-	
<b>Newly Added Requirements</b>			

### 3. Updated Class Diagram

Since the design is changed from the beginning to improve the relations between classes, the Class Diagram of the application is changed.



### 4. Explanation of What is Being Done and What has been Done How

#### 4.1. Security Configuration- [Link](#)

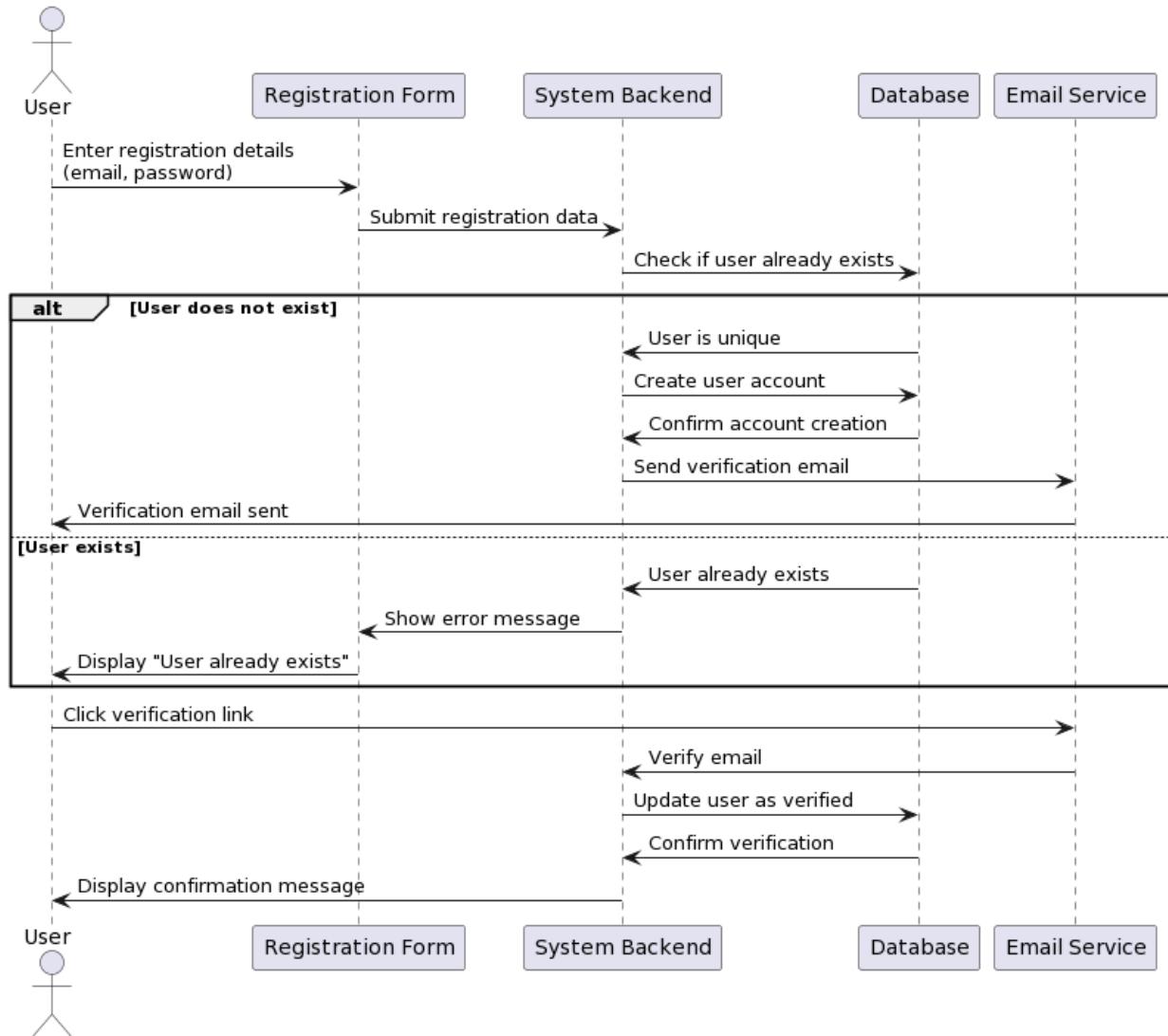
1. Spring Security allows all of the users to get to the “register”, “login”, and “index” pages.
2. Other than these pages, authentication is needed.
3. For logging in, it directs users to the login page
4. Default for a successful login directs users to home page where show communities and create community buttons are displayed.

## 4.2. User Registration

User Registration is done as shown in my previous Sequence Diagram. Here is a breakdown of how my design works:

1. User clicks /Register
2. Auth Controller Controls the /Register and shows a Model to get necessary attributes to create a User
3. From the model created by Thymeleaf CustomUserDetailsService takes the necessary attributes and sets it as a user
4. CustomUserDetailsService save user to the UserRepository

[CustomUserDetailsService](#) - [AuthController](#) – [login.html](#) – [register.html](#) – [hello.html](#)



## 4.3. Basic Structure of the Project

### Models - [Link](#)

This is where our classes are. In here, we automatically create tables by using hibernate (update every time so no data loss occurs when redeployed). Also in here, we define the relationships between the classes with Java hibernate annotations such as (@OneToMany, @ManyToOne, @Column (nullable = true ), @Column(nullable = false) etc...).

### Controller- [Link](#)

This is where we control our applications' urls. From here, with @GetMapping annotation we create models and tell the spring what we want to take as attributes then redirect it to a Forms. In addition to that, by using @PostMapping, we get the input of the data entered into the Forms and start services according to the data obtained to post data into our database.

### Service - [Link](#)

This is where our main operation runs. All of the necessary operations are done here. This level is where our data enters and being rendered. Such as saveuser, authenticateuser, savecommunity. This is basically operation and method level.

### Repository- [Link](#)

This is where our database and application get together. By implementing Spring JPA, we can define methods such as findById defaultly from our database to get data. This part is basically used to obtain Data from database and save Data to database

### Dto - [Link](#)

Data transfer objects are used to transfer data from models to controllers and services they serve us as a waitress, some function orders for Dto and it gets it from a model, controller or service and transfers the data.

### resources/Templates - [Link](#)

Templates are the views, (frontends) of our application. In here, I used Thymeleaf to bind the frontend with backend.

## 4.4. Issues

### 4.4.1. Open Issues - [Link](#)

Issues are created for the project so that the scope of the project is more absolute and more visible to the eye.

7 Open ✓ 18 Closed		Author ▾	Label ▾	Projects ▾	Milestones ▾	Assignee ▾	Sort ▾
<a href="#">Discussion -(Comment, upVote, etc.) Creation</a>	Coding						
#27	opened 4 hours ago by AhmetZ06	Final Deliverable					
<a href="#">User Profiles Creation</a>	Coding						
#26	opened 4 hours ago by AhmetZ06						
<a href="#">Post Template Creation</a>	Coding						
#25	opened 4 hours ago by AhmetZ06						
<a href="#">Double UserID creation by Spring JPA</a>	bug						
#24	opened 4 hours ago by AhmetZ06	Final Deliverable					
<a href="#">When Private Community is tried to be entered, assign the moderators and admins of the related community a task - accept or decline</a>	Coding enhancement						
#23	opened yesterday by AhmetZ06	Final Deliverable					
<a href="#">Deploy Application</a>	Coding enhancement						
#22	opened yesterday by AhmetZ06	22 april 17:00 -- ...					
<a href="#">Create Post Function</a>	Coding						
#18	opened 2 days ago by AhmetZ06	22 april 17:00 -- ...					

#### 4.4.2. Closed Issues - [Link](#)

Issues are closed when a certain task is completed or when a bug or a problem is solved

Author ▾ Label ▾ Projects ▾ Milestones ▾ Assignee ▾ Sort ▾	
7 Open	18 Closed
>Create Generic Template For each Community	Coding
#21 by AhmetZ06 was closed 4 hours ago	22 april 17:00 -- ...
>Show Every Available Communities	Coding
#20 by AhmetZ06 was closed yesterday	22 april 17:00 -- ...
Create Default Post Template	Coding
#19 by AhmetZ06 was closed 4 hours ago	22 april 17:00 -- ...
Creating a Community With Default Admin	bug enhancement
#17 by AhmetZ06 was closed 4 hours ago	
505 error when trying to create communities	bug
#16 by AhmetZ06 was closed 2 days ago	22 april 17:00 -- ...
Security Config	bug
#15 by AhmetZ06 was closed last week	
Add the Project to Git	Coding
#13 by AhmetZ06 was closed 2 weeks ago	
Create Classes	Coding
#12 by AhmetZ06 was closed last week	7 tasks done
Connection to a DataBase	Coding
#11 by AhmetZ06 was closed last week	1 task done
Project Plan Creation	documentation
#10 by AhmetZ06 was closed on Mar 24	Milestone with d...
Design Diagrams for the Project - UML, Use Case etc.	documentation
#8 by AhmetZ06 was closed on Mar 24	3 tasks done
MockUp Scenarios and Scenes	enhancement
#7 by AhmetZ06 was closed on Mar 12	9 tasks done
Create a SRS Document	documentation
#6 by AhmetZ06 was closed on Mar 4	5 tasks done
Requirement Elicitation	enhancement
#5 by AhmetZ06 was closed on Feb 26	5 tasks done
Join Discord	good first issue
#4 by AhmetZ06 was closed on Feb 25	
First Week's Homework - Redo	documentation
#3 by AhmetZ06 was closed on Feb 25	5 tasks done
First Week Homework	enhancement
#2 by AhmetZ06 was closed on Feb 25	
Learn How to Create a Web Application ?	good first issue
#1 by AhmetZ06 was closed on Feb 19	

## 4.5. Documentation

A complete [SRS](#) document is presented at the Wiki page. After the project is complete, it will be evaluated and updated. For the design of the system, the diagrams shown in the [wiki](#) home page are used. After the project is completed, additional diagrams will be added to make the project more understandable.

## 4.6. Research and Management Decisions

While doing the project, of course there has been a lot of research about how to do a project. While doing so and learning about Spring Boot, there has been a confrontation with Spring MVC. After that, for the goal of simplifying the project, I decided to move the project from Spring Boot to Spring MVC since the frontend application of it is more easier, (API approach methods and controllers also made up to some point 😊 ).

## 4.7. References

1. <https://www.baeldung.com/registration-with-spring-mvc-and-spring-security>
2. Udemy Courses About Spring Boot and Spring MVC
3. Youtube Courses About Spring Boot and Spring MVC
4. <https://spring.io/>
5. <https://docs.spring.io/spring-framework/reference/index.html>
6. ChatGPT for debugging purposes